Finance Pillow

CREATE DOMAIN email AS VARCHAR(100)

CHECK(VALUE ~ '^\w+@[a-zA-Z\_]+?\.[a-zA-Z]{2,3}$');

create table users(

user\_id int PRIMARY KEY,

username varchar (255) NOT NULL,

email email UNIQUE NOT NULL,

user\_password varchar (255) NOT NULL

);

create table incomes(

income\_sum decimal(10,2) NOT null,

user\_id int,

FOREIGN KEY(user\_id)

REFERENCES Users(user\_id)

);

create table category(

category\_id int PRIMARY KEY,

name\_category varchar(255),

user\_id int,

FOREIGN KEY(user\_id)

REFERENCES Users(user\_id)

);

create table expenses(

expenses\_sum decimal(10,2) NOT null,

user\_id int,

category\_id int,

FOREIGN KEY(user\_id)

REFERENCES Users(user\_id),

FOREIGN KEY(category\_id)

REFERENCES Category(category\_id)

);

create table sum\_category(

sum\_for\_category decimal(10,2) NOT null,

user\_id int,

category\_id int,

FOREIGN KEY(user\_id)

REFERENCES Users(user\_id),

FOREIGN KEY(category\_id)

REFERENCES Category(category\_id)

);

INSERT INTO users (user\_id, username, email, user\_password)

SELECT

i AS user\_id,

'user' || i AS username,

'user' || i || '@gmail.com' AS email,

'password' || i AS user\_password

FROM generate\_series(1, 30) AS i;

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 1);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 2);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 3);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 4);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 5);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 6);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 7);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 8);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 9);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 10);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 11);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 12);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 13);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 14);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 15);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 16);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 17);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 18);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 19);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 20);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 21);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 22);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 23);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 24);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 25);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 26);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 27);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 28);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 29);

INSERT INTO incomes (income\_sum, user\_id) VALUES (FLOOR(RANDOM() \* 1000 + 1), 30);

SET @user\_id := 1;

-- Почніть цикл для вставки даних

WHILE @user\_id <= 30 DO

INSERT INTO expenses (expenses\_sum, user\_id) VALUES (FLOOR(RAND() \* 1000 + 1), @user\_id);

SET @user\_id := @user\_id + 1;

END WHILE;

INSERT INTO category (category\_id, name\_category, user\_id)

SELECT

i AS category\_id,

'Category' || i AS name\_category,

i AS user\_id

FROM generate\_series(1, 30) AS i;

-- Заповнення таблиці "expenses" рандомними записами з використанням зовнішніх ключів

INSERT INTO expenses (expenses\_sum, user\_id, category\_id)

SELECT

(random() \* 1000)::decimal(10, 2) as expenses\_sum,

(floor(random() \* 30 + 1))::int as user\_id,

(floor(random() \* 30 + 1))::int as category\_id

FROM generate\_series(1, 30) AS i

WHERE (random() \* 30 + 1) IS NOT NULL;

-- Заповнення таблиці "sum\_category" рандомними записами з використанням зовнішніх ключів

INSERT INTO sum\_category (sum\_for\_category, user\_id, category\_id)

SELECT

(random() \* 1000)::decimal(10, 2) as sum\_for\_category,

(floor(random() \* 30 + 1))::int as user\_id,

(floor(random() \* 30 + 1))::int as category\_id

FROM generate\_series(1, 30) AS i

WHERE

(floor(random() \* 30 + 1)) IS NOT NULL AND

(floor(random() \* 30 + 1)) IS NOT NULL;